

On Failing Sets of the Interval-Passing Algorithm for Compressed Sensing

Yauhen Yakimenka[†] and Eirik Rosnes[‡]

[†]Institute of Computer Science, University of Tartu, Tartu 50409, Estonia

Email: yauhen@ut.ee

[‡]Department of Informatics, University of Bergen, N-5020 Bergen, Norway, and the Simula Research Lab

Email: eirik@ii.uib.no

Abstract—In this work, we analyze the failing sets of the interval-passing algorithm (IPA) for compressed sensing. The IPA is an efficient iterative algorithm for reconstructing a k -sparse nonnegative n -dimensional real signal x from a small number of linear measurements y . In particular, we show that the IPA fails to recover x from y if and only if it fails to recover a corresponding binary vector of the same support, and also that only positions of nonzero values in the measurement matrix are of importance for success of recovery. Based on this observation, we introduce *termatiko sets* and show that the IPA fails to fully recover x if and only if the support of x contains a nonempty *termatiko set*, thus giving a complete (graph-theoretic) description of the failing sets of the IPA. Finally, we present an extensive numerical study showing that in many cases there exist *termatiko sets* of size strictly smaller than the stopping distance of the binary measurement matrix; even as low as half the stopping distance in some cases.

I. INTRODUCTION

THE reconstruction of a (mathematical) object from a partial set of observations in an efficient and reliable manner is of fundamental importance. Compressed sensing, motivated by the ground-breaking work of Candès and Tao [1], [2], and independently by Donoho [3], is a research area in which the object to be reconstructed is a k -sparse signal vector (there are at most k nonzero entries in the vector) over the real numbers. The partial information provided is a linear transformation of the signal vector, the *measurement vector*, and the objective is to reconstruct the object from a small number of measurements. Compressed sensing provides a mathematical framework which shows that, under some conditions, signals can be recovered from far less measurements than with conventional signal acquisition methods. The main idea in compressed sensing is to exploit that most interesting signals have an inherent structure or contain redundancy.

Iterative reconstruction algorithms for compressed sensing have received considerable interest recently. See, for instance, [4]–[8] and references therein. The interval-passing algorithm (IPA) for reconstruction of nonnegative sparse signals was

introduced by Chandar *et al.* in [6] for binary measurement matrices. The algorithm was further generalized to nonnegative real measurement matrices in [5].

In this work, we show that the IPA fails for a nonnegative signal $x = (x_1, \dots, x_n) \in \mathbb{R}_{\geq 0}^n$, $\mathbb{R}_{\geq 0}$ is the set of nonnegative real numbers, if and only if it fails for a corresponding binary vector z of the same support, and also that only positions of nonzero values in the measurement matrix are of importance for success of recovery. Thus, failing sets as subsets of $[n] \triangleq \{1, \dots, n\}$ can be defined. It has previously been shown that traditional stopping sets for belief propagation decoding of low-density parity-check (LDPC) codes are failing sets of the IPA, in the sense that if the support of a signal $x \in \mathbb{R}_{\geq 0}^n$ contains a nonempty stopping set, then the IPA fails to fully recover x [5, Thm. 1]. In this work, we extend the results in [5] and define *termatiko sets* (which contain stopping sets as a special case) and show that the IPA fails to fully recover a signal $x \in \mathbb{R}_{\geq 0}^n$ if and only if the support of x contains a nonempty *termatiko set*, thus giving a complete (graph-theoretic) description of the failing sets of the IPA. Finally, we present an extensive numerical study which includes both specific binary parity-check matrices of LDPC codes and parity-check matrices from LDPC code ensembles as measurement matrices. The numerical results show that in many cases there exist *termatiko sets* of size strictly smaller than the stopping distance of the measurement matrix; even as low as half the stopping distance in some cases, where the stopping distance s_{\min} of a measurement matrix is the minimum size of a nonempty stopping set of the matrix when it is regarded as a parity-check matrix of a binary LDPC code.

We remark that the performance of the IPA and its comparison with other algorithms for efficient reconstruction of sparse signals have been investigated in [5] (see Figs. 4 and 8), and we refer the interested reader to that work for such results.

II. NOTATION AND BACKGROUND

In this section, we introduce the problem formulation, revise notation from [5], and describe the IPA in detail.

A. Compressed Sensing

Let $x \in \mathbb{R}^n$, where \mathbb{R} is the field of real numbers, be an n -dimensional k -sparse signal (i.e., it has at most k nonzero

The work of Y. Yakimenka and E. Rosnes was partially funded by the Norwegian-Estonian Research Cooperation Programme (grant EMP133). The work of E. Rosnes was partially funded by the Research Council of Norway (grant 240985/F20) and by Simula@UiB.

The calculations were carried out in part in the High Performance Computing Centre of University of Tartu.

entries), and let $A = (a_{ji})$ be an $m \times n$ real measurement matrix. We consider the recovery of \mathbf{x} from measurements $\mathbf{y} = A\mathbf{x} \in \mathbb{R}^m$, where $m < n$ and $k < n$.

The reconstruction problem of compressed sensing is to find the sparsest \mathbf{x} (or the one that minimizes the ℓ_0 -norm) under the constraint $\mathbf{y} = A\mathbf{x}$, which in general is an NP-hard problem. Basis pursuit is an algorithm which reconstructs \mathbf{x} by minimizing its ℓ_1 -norm under the constraint $\mathbf{y} = A\mathbf{x}$ [2]. This is a linear program, and thus it can be solved in polynomial time. The algorithm has a remarkable performance, but its complexity is high, making it impractical for many applications that require fast reconstruction. A fast reconstruction algorithm for nonnegative real signals and measurement matrices is the IPA which is described below in Section II-C.

B. Tanner Graph Representation

We associate with matrix A the bipartite Tanner graph $G = (V \cup C, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is a set of *variable nodes*, $C = \{c_1, c_2, \dots, c_m\}$ is a set of *measurement nodes*, and E is a set of edges from C to V . We will often equate V with $[n]$ and C with $[m]$. There is an edge in E between $c \in C$ and $v \in V$ if and only if $a_{cv} \neq 0$. We also denote the sets of neighbors for each node $v \in V$ and $c \in C$ as

$$\mathcal{N}(v) = \{c \in C \mid (c, v) \in E\}$$

$$\text{and } \mathcal{N}(c) = \{v \in V \mid (c, v) \in E\},$$

respectively. Furthermore, if $T \subset V$ or $T \subset C$ and $w \in V \cup C$, then define

$$\mathcal{N}(T) = \bigcup_{t \in T} \mathcal{N}(t) \text{ and } \mathcal{N}_T(w) = \mathcal{N}(w) \cap T.$$

A stopping set [9] of the Tanner graph G is defined as a subset S of V such that all its neighboring measurement nodes are connected to S at least twice.

C. Interval-Passing Algorithm

The IPA is an iterative algorithm to reconstruct a nonnegative real signal $\mathbf{x} \in \mathbb{R}_{\geq 0}^n$ from a set of linear measurements $\mathbf{y} = A\mathbf{x}$, introduced by Chandar *et al.* in [6] for binary measurement matrices. The algorithm was extended to nonnegative real measurement matrices in [5], and this is the case that we will consider. The IPA iteratively sends messages between variable and measurement nodes. Each message contains two real numbers, a *lower bound* and an *upper bound* on the value of the variable node to which it is affiliated. Let $\mu_{v \rightarrow c}^{(\ell)}$ (resp. $\mu_{c \rightarrow v}^{(\ell)}$) denote the lower bound of the message from variable node v (resp. measurement node c) to measurement node c (resp. variable node v) at iteration ℓ . The corresponding upper bound of the message is denoted by $M_{v \rightarrow c}^{(\ell)}$ (resp. $M_{c \rightarrow v}^{(\ell)}$). It is a distinct property of the algorithm that at any iteration ℓ , $\mu_{v \rightarrow c}^{(\ell)} \leq x_v \leq M_{v \rightarrow c}^{(\ell)}$ and $\mu_{c \rightarrow v}^{(\ell)} \leq x_v \leq M_{c \rightarrow v}^{(\ell)}$, for all $v \in V$ and $c \in \mathcal{N}(v)$.

The detailed steps of the IPA are shown in Algorithm 1 below. From Lines 3, 16, and 17 one can see that both $\mu_{v \rightarrow c}^{(\ell)}$ and $M_{v \rightarrow c}^{(\ell)}$ are independent of $c \in \mathcal{N}(v)$. Thus, we will occasionally denote $\mu_{v \rightarrow c}^{(\ell)}$ by $\mu_{v \rightarrow \cdot}^{(\ell)}$ and $M_{v \rightarrow c}^{(\ell)}$ by $M_{v \rightarrow \cdot}^{(\ell)}$.

Algorithm 1 Interval-Passing Algorithm (cf. [5, Alg. 1])

```

1: function IPA( $\mathbf{y}, A$ )
   Initialization
2:   for all  $v \in V, c \in \mathcal{N}(v)$  do
3:      $\mu_{v \rightarrow c}^{(0)} \leftarrow 0$  and  $M_{v \rightarrow c}^{(0)} \leftarrow \min_{c' \in \mathcal{N}(v)} (y_{c'} / a_{c'v})$ 
4:   end for
   Iterations
5:    $\ell \leftarrow 0$ 
6:   repeat
7:      $\ell \leftarrow \ell + 1$ 
8:     for all  $c \in C, v \in \mathcal{N}(c)$  do
9:        $\mu_{c \rightarrow v}^{(\ell)} \leftarrow \frac{1}{a_{cv}} \left( y_c - \sum_{v' \in \mathcal{N}(c), v' \neq v} a_{cv'} M_{v' \rightarrow c}^{(\ell-1)} \right)$ 
10:      if  $\mu_{c \rightarrow v}^{(\ell)} < 0$  then
11:         $\mu_{c \rightarrow v}^{(\ell)} \leftarrow 0$ 
12:      end if
13:       $M_{c \rightarrow v}^{(\ell)} \leftarrow \frac{1}{a_{cv}} \left( y_c - \sum_{v' \in \mathcal{N}(c), v' \neq v} a_{cv'} \mu_{v' \rightarrow c}^{(\ell-1)} \right)$ 
14:    end for
15:    for all  $v \in V, c \in \mathcal{N}(v)$  do
16:       $\mu_{v \rightarrow c}^{(\ell)} \leftarrow \max_{c' \in \mathcal{N}(v)} \mu_{c' \rightarrow v}^{(\ell)}$ 
17:       $M_{v \rightarrow c}^{(\ell)} \leftarrow \min_{c' \in \mathcal{N}(v)} M_{c' \rightarrow v}^{(\ell)}$ 
18:    end for
19:  until  $\mu_{v \rightarrow \cdot}^{(\ell)} = \mu_{v \rightarrow \cdot}^{(\ell-1)}$  and  $M_{v \rightarrow \cdot}^{(\ell)} = M_{v \rightarrow \cdot}^{(\ell-1)}, \forall v \in V$ 
  Result
20:  for all  $v \in V$  do  $\hat{x}_v \leftarrow \mu_{v \rightarrow \cdot}^{(\ell)}$  end for
21:  return  $\hat{\mathbf{x}}$ 
22: end function

```

III. FAILING SETS OF THE INTERVAL-PASSING ALGORITHM

In this section, we present several results related to the failure of the IPA. In particular, in Section III-A, we show that the IPA fails to recover \mathbf{x} from \mathbf{y} if and only if it fails to recover a corresponding binary vector of the same support, and also that only positions of nonzero values in the matrix A are of importance for success of recovery (see Lemma 1 below). Based on Lemma 1, we introduce the concept of *termatiko* sets in Section III-B and give a complete (graph-theoretic) description of the failing sets of the IPA in Section III-C.

A. Signal Support Recovery

Consider the two related problems IPA(\mathbf{y}, A) and IPA(\mathbf{s}, B), where $\mathbf{s} = B\mathbf{z}$ and $\mathbf{z} \in \{0, 1\}^n$ has support $\text{supp}(\mathbf{z}) = \text{supp}(\mathbf{x})$, i.e., \mathbf{x} and \mathbf{z} have the same support. The support of a real vector $\mathbf{x} \in \mathbb{R}^n$ is defined as the set of nonzero coordinates of \mathbf{x} . The binary matrix B contains ones exactly in the positions where A has nonzero values. We will show below (see Lemma 1) that these two problems behave identically, namely they recover exactly the same positions of \mathbf{x} and \mathbf{z} . However, note that this is true if the identical algorithm

(Algorithm 1) is applied to both problems, i.e., the binary nature of z is not exploited.

Lemma 1. Let $A = (a_{ji}) \in \mathbb{R}_{\geq 0}^{m \times n}$, $\mathbf{x} \in \mathbb{R}_{\geq 0}^n$, $B = (b_{ji}) \in \{0, 1\}^{m \times n}$, and $\mathbf{z} \in \{0, 1\}^n$, where $\text{supp}(\mathbf{z}) = \text{supp}(\mathbf{x})$ and

$$b_{ji} = \begin{cases} 0 & \text{if } a_{ji} = 0, \\ 1 & \text{otherwise.} \end{cases}$$

Further, denote $\mathbf{y} = A\mathbf{x}$, $\mathbf{s} = B\mathbf{z}$, $\hat{\mathbf{x}} = \text{IPA}(\mathbf{y}, A)$, and $\hat{\mathbf{z}} = \text{IPA}(\mathbf{s}, B)$. Then, for all $v \in V$,

$$\hat{x}_v = x_v \text{ if and only if } \hat{z}_v = z_v.$$

Proof: Define subsets of V in which either the lower or the upper bound of a variable-to-measurement message, at a given iteration ℓ , is equal to x_v or z_v as follows:

$$\begin{aligned} \gamma_x^{(\ell)} &= \{v \in V \mid \mu_{v \rightarrow \cdot}^{(\ell)} = x_v\}, \Gamma_x^{(\ell)} = \{v \in V \mid M_{v \rightarrow \cdot}^{(\ell)} = x_v\}, \\ \gamma_z^{(\ell)} &= \{v \in V \mid \lambda_{v \rightarrow \cdot}^{(\ell)} = z_v\}, \Gamma_z^{(\ell)} = \{v \in V \mid \Lambda_{v \rightarrow \cdot}^{(\ell)} = z_v\}, \end{aligned}$$

where $\lambda_{v \rightarrow \cdot}^{(\ell)}$ and $\Lambda_{v \rightarrow \cdot}^{(\ell)}$ denote, respectively, the lower and the upper bound of the variable-to-measurement message from variable node v to any measurement node $c \in \mathcal{N}(v)$ at iteration ℓ for $\text{IPA}(\mathbf{s}, B)$ (analogously to $\mu_{v \rightarrow \cdot}^{(\ell)}$ and $M_{v \rightarrow \cdot}^{(\ell)}$ for $\text{IPA}(\mathbf{y}, A)$).

To prove the lemma, it is enough to show that at each iteration ℓ , $\gamma_x^{(\ell)} = \gamma_z^{(\ell)}$ and $\Gamma_x^{(\ell)} = \Gamma_z^{(\ell)}$. We demonstrate this by induction on ℓ .

Base Case.

$$\begin{aligned} \gamma_x^{(0)} &= \{v \in V \mid x_v = 0\} = \{v \in V \mid z_v = 0\} = \gamma_z^{(0)}, \\ \Gamma_x^{(0)} &= \{v \in V \mid \exists c \in \mathcal{N}(v), \text{ s.t. } y_c = a_{cv}x_v\} \\ &= \{v \in V \mid \exists c \in \mathcal{N}(v), \text{ s.t. } s_c = z_v\} = \Gamma_z^{(0)}. \end{aligned}$$

Inductive Step.

Consider iteration $\ell \geq 1$. First note that all $v \in V$ with $x_v = 0$ (and hence $z_v = 0$) belong to both $\gamma_x^{(\ell)}$ and $\gamma_z^{(\ell)}$.

If $x_v > 0$ (and hence $z_v = 1$) then from Line 16 of Algorithm 1 and the definition of $\gamma_x^{(\ell)}$, we have $v \in \gamma_x^{(\ell)}$ if and only if there exists $c \in \mathcal{N}(v)$ such that $\mu_{c \rightarrow v}^{(\ell)} = x_v$. More precisely:

$$\begin{aligned} a_{cv}x_v &= y_c - \sum_{\substack{v' \in \mathcal{N}(c) \\ v' \neq v}} a_{cv'} M_{v' \rightarrow c}^{(\ell-1)} \\ &= a_{cv}x_v + \sum_{\substack{v' \in \mathcal{N}(c) \\ v' \neq v}} a_{cv'} (x_{v'} - M_{v' \rightarrow c}^{(\ell-1)}) \leq a_{cv}x_v. \end{aligned}$$

Equality holds if and only if $M_{v' \rightarrow c}^{(\ell-1)} = x_{v'}$ for all $v' \in \mathcal{N}(c) \setminus \{v\}$ or, in our notation, $\mathcal{N}(c) \setminus \{v\} \subset \Gamma_x^{(\ell-1)}$. However, by inductive assumption $\Gamma_z^{(\ell-1)} = \Gamma_x^{(\ell-1)}$ and hence $\Lambda_{v' \rightarrow c}^{(\ell-1)} = z_{v'}$ for all $v' \in \mathcal{N}(c) \setminus \{v\}$. This is equivalent to $\lambda_{c \rightarrow v}^{(\ell)} = z_v$ and thus $v \in \gamma_z^{(\ell)}$.

Hence, for all $v \in V$, v either belongs to both $\gamma_x^{(\ell)}$ and $\gamma_z^{(\ell)}$, or to none of them.

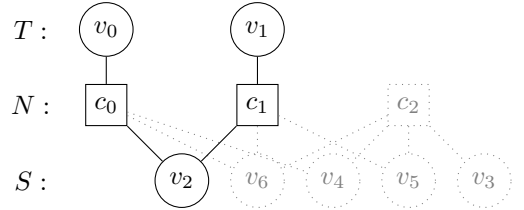


Fig. 1. Example of a termatiko set T with all measurement nodes in N connected to both T and S (cf. Theorem 1). The rest of the Tanner graph is drawn dotted.

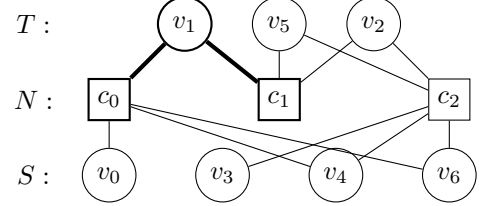


Fig. 2. Example of a termatiko set T with a measurement node c_1 connected to T only (cf. Theorem 1). Highlighted is the connection to a measurement node c_0 , which is connected to T only once.

Analogously, we can show that $\Gamma_x^{(\ell)} = \Gamma_z^{(\ell)}$. Details are omitted for brevity. ■

Lemma 1 gives a powerful tool for analysis of IPA performance. Instead of considering $A \in \mathbb{R}_{\geq 0}^{m \times n}$ and $\mathbf{x} \in \mathbb{R}_{\geq 0}^n$ we need only to work with binary A and \mathbf{x} (although all operations are still performed over \mathbb{R}). Thus, in the rest of the paper, we assume that A is binary.

B. Termatiko Sets

We define termatiko sets through failures of the IPA.

Definition 1. We call $T \subset V$ a termatiko set if and only if $\text{IPA}(A\mathbf{x}_T, A) = \mathbf{0}$, where \mathbf{x}_T is a binary vector with support $\text{supp}(\mathbf{x}_T) = T$.

From Lemma 1, it follows that the IPA completely fails to recover $\mathbf{x} \in \mathbb{R}_{\geq 0}^n$ if and only if $\text{supp}(\mathbf{x}) = T$, where T is a nonempty termatiko set.

Theorem 1. Let T be a subset of the set of variable nodes V . We denote by $N = \mathcal{N}(T)$ the set of measurement nodes connected to T and also denote by S the other variable nodes connected only to N as follows:

$$S = \{v \in V \setminus T : \mathcal{N}_N(v) = \mathcal{N}(v)\}.$$

Then, T is a termatiko set if and only if for each $c \in N$ one of the following two conditions holds (cf. Figs. 1 and 2):

- c is connected to S (this implies $S \neq \emptyset$);
- c is not connected to S and

$$\left| \{v \in \mathcal{N}_T(c) : \forall c' \in \mathcal{N}(v), |\mathcal{N}_T(c')| \geq 2\} \right| \geq 2.$$

Proof: Consider the problem $\text{IPA}(A\mathbf{x}_T, A)$, where \mathbf{x}_T is a binary vector with support $\text{supp}(\mathbf{x}_T) = T$.

We first note that measurement nodes in $C \setminus N$ have value zero and hence all variable nodes connected to them (i.e., $v \in V \setminus (T \cup S)$) are recovered with zeros at the initialization step of Algorithm 1. As a consequence, they can be safely pruned and w.l.o.g. we can assume that $C = N$ and $V = T \cup S$.

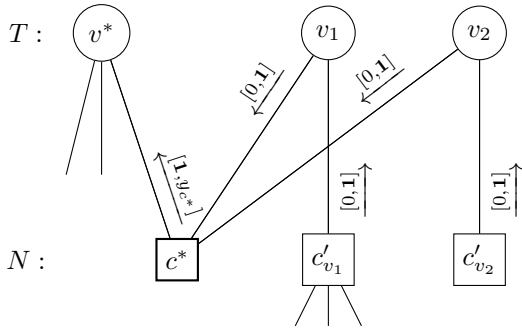


Fig. 3. Exact bounds propagation in a nontermatiko set. Here $[\mu, M]$ denotes sending a lower bound of μ and an upper bound of M in the direction given by the corresponding arrow. Numbers in bold are exact bounds.

Assume that T satisfies the conditions of the theorem and consider the problem $\text{IPA}(Ax_T, A)$. We show by induction that for all $v \in T \cup S$ at each iteration $\ell \geq 0$ it holds that $\mu_{v \rightarrow}^{(\ell)} = 0$ and $M_{v \rightarrow}^{(\ell)} \geq 1$. Moreover, each measurement node $c \in N$ that is not connected to S has at least two different neighbors $v_1, v_2 \in T$ with $M_{v_1 \rightarrow}^{(\ell)} \geq 2$ and $M_{v_2 \rightarrow}^{(\ell)} \geq 2$.

Base Case.

For $\ell = 0$ we immediately obtain from Algorithm 1 that $\mu_{v \rightarrow}^{(0)} = 0$ and, as each $c \in N$ has at least one nonzero neighbor, $M_{v \rightarrow}^{(0)} \geq 1$. In addition, consider $c \in N$ that is not connected to S . It has at least two different neighbors $v_1, v_2 \in T$, each connected only to measurement nodes with not less than two neighbors in T . Therefore, $M_{v_1 \rightarrow} \geq 2$ and $M_{v_2 \rightarrow} \geq 2$.

Inductive Step.

Consider $\ell \geq 1$. For all $c \in N$ and all $v \in \mathcal{N}(c)$,

$$M_{c \rightarrow v}^{(\ell)} = y_c - \sum_{v' \in \mathcal{N}(c), v' \neq v} \mu_{v' \rightarrow c}^{(\ell-1)} = y_c.$$

Hence, upper bounds are exactly the same as for $\ell = 0$ and the same inequalities hold for them.

In order to find lower bounds, we consider two cases for $c \in N$. If c is connected to S , then

$$y_c - \sum_{\substack{v' \in \mathcal{N}(c) \\ v' \neq v}} M_{v' \rightarrow c}^{(\ell-1)} \leq (|\mathcal{N}(c)| - 1) - \sum_{\substack{v' \in \mathcal{N}(c) \\ v' \neq v}} 1 = 0$$

and therefore $\mu_{c \rightarrow v}^{(\ell)} = 0$. If c is connected to T only, then

$$y_c - \sum_{\substack{v' \in \mathcal{N}(c) \\ v' \neq v}} M_{v' \rightarrow c}^{(\ell-1)} \leq |\mathcal{N}_T(c)| - \left(1 + \sum_{\substack{v' \in \mathcal{N}_T(c) \\ v' \neq v}} 1\right) = 0$$

and again $\mu_{c \rightarrow v}^{(\ell)} = 0$. Here, the extra 1 inside the parenthesis indicates the fact that for at least one v' we have $M_{v' \rightarrow c}^{(\ell-1)} \geq 2$. Thus, at each iteration of the IPA for each $v \in V$ the lower bound is equal to zero, and the algorithm will return $\hat{x} = 0$.

We have demonstrated that if T satisfies the conditions of the theorem, it is a termatiko set. What remains to be proven is that if T does not satisfy the conditions of the theorem, the IPA can recover at least some of the nonzero values.

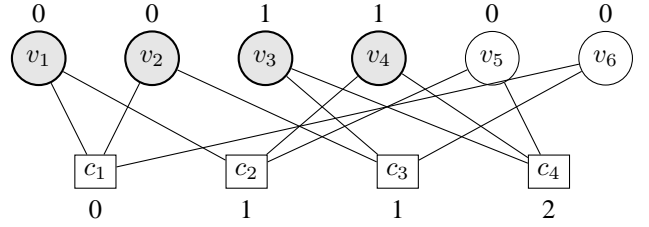


Fig. 4. Counter-example to [5, Thm. 2]. The set of variable nodes is $V = \{v_1, \dots, v_6\}$ (circles) and the set of measurement nodes is $C = \{c_1, \dots, c_4\}$ (squares). The integer attached to a node is its corresponding value (x_{v_i} for variable node v_i and y_{c_i} for measurement node c_i). $V_S = \{v_1, v_2, v_3, v_4\} \subset V$ (shaded in gray) is a minimal stopping set and c_1 is a zero-valued ($y_{c_1} = 0$) measurement node connected to V_S . Note that v_5 is not in V_S , but exactly because of it, the IPA cannot correctly recover v_4 .

Assume that there exists $c^* \in N$ not connected to S (i.e., $\mathcal{N}_T(c^*) = \mathcal{N}(c^*)$) and such that

$$\left| \{v \in \mathcal{N}_T(c^*) : \forall c' \in \mathcal{N}(v), |\mathcal{N}_T(c')| \geq 2\} \right| \leq 1.$$

If this set has one element, denote it by v^* . If it is empty, let v^* be any element of $\mathcal{N}_T(c^*)$.

A special case when $|\mathcal{N}_T(c^*)| = 1$ is trivial. Otherwise, for any $v \in \mathcal{N}_T(c^*) \setminus \{v^*\}$, there exists $c'_v \in \mathcal{N}(v)$ such that $|\mathcal{N}_T(c'_v)| \leq 1$, which in truth means that $\mathcal{N}_T(c'_v) = \{v\}$.

Hence, at the initialization step of the IPA, for all $v \in \mathcal{N}_T(c^*) \setminus \{v^*\}$ we will have $\mu_{v \rightarrow}^{(0)} = 0$ and $M_{v \rightarrow}^{(0)} = 1$. Therefore, at iteration $\ell = 1$:

$$\mu_{c^* \rightarrow v^*}^{(1)} \leftarrow y_{c^*} - \sum_{\substack{v \in \mathcal{N}_T(c^*) \\ v \neq v^*}} M_{v \rightarrow c^*}^{(0)} = y_{c^*} - \sum_{\substack{v \in \mathcal{N}_T(c^*) \\ v \neq v^*}} 1 = 1.$$

Thus, the IPA will output 1 for position $v^* \in T$, which means that T is not a termatiko set. See Fig. 3 for illustration. ■

Theorem 1 gives a precise graph-theoretic description of termatiko sets. In fact, it defines two important subclasses of termatiko sets; stopping sets and sets with all $c \in N$ connected to both T and S . Also, it is worth noting that $T \cup S$ is a stopping set. Thus, a termatiko set is always a subset of some stopping set. We define the size of the smallest nonempty termatiko set as the *termatiko distance*.

C. General Failing Sets

In Section III-B, we defined termatiko sets as supports of binary vectors that avert the IPA from recovering any of the ones. However, the algorithm can recover only some of the positions of ones. The next proposition (proof omitted) gives a connection between (partial) failures of the IPA and termatiko sets; it shows that the IPA fails on any signal in $\mathbb{R}_{\geq 0}^n$ if and only if its support contains a nonempty termatiko set.

Proposition 1. *The IPA fails to fully recover a nonnegative real signal $x \in \mathbb{R}_{\geq 0}^n$ if and only if the support of x contains a nonempty termatiko set.*

IV. COUNTER-EXAMPLE TO [5, THM. 2]

In [5, Thm. 2], a condition for full recovery of x is stated. However, in Fig. 4, we provide a counter-example to this theorem. Note that the Tanner graph of Fig. 4 is $(2, 3)$ -regular (only regular Tanner graphs with variable node degree at least

TABLE I

ESTIMATED TERMATIKO SET SIZE SPECTRA (INITIAL PART) OF MEASUREMENT MATRICES FROM SECTION VI, WHERE \hat{h}_{\min} DENOTES THE ESTIMATED TERMATIKO DISTANCE. \mathfrak{T}_1 CORRESPONDS TO TERMATIKO SETS WITH ALL MEASUREMENT NODES IN N CONNECTED TO BOTH T AND S , AND \mathfrak{T}_2 CORRESPONDS TO ALL THE REMAINING TERMATIKO SETS. ALSO SHOWN ARE THE EXACT STOPPING DISTANCES AND STOPPING SET SIZE SPECTRA (INITIAL PART). ENTRIES IN BOLD ARE EXACT VALUES. FOR $A^{(1)}$, THE HEURISTIC APPROACH GIVES A MULTIPLICITY OF 5875518 FOR SIZE 5, WHILE THE EXACT NUMBER IS 6318378 (AN UNDERESTIMATION OF ABOUT 7.5%).

Measurement matrix	\hat{h}_{\min}	Initial estimated termatiko set size spectrum	s_{\min}	Initial stopping set size spectrum
$A^{(1)}$	3	\mathfrak{T}_1 : (3630, 93775, 6318378 , 48548225, 71709440, 36514170, 7969060, 856801, 41745) \mathfrak{T}_2 : (0, 0, 0, 410190, 18610405, 71153445, 86844725, 58849681, 28430160)	6	(1815, 605, 45375, 131890, 3550382, 28471905)
$A^{(2)}$	9	\mathfrak{T}_1 : (465, 3906, 12555, 8835, 0, 0, ...) \mathfrak{T}_2 : (0, 0, 0, 1860, 5115, 10695, 2325, 5580, 2325, 6045, 10850, 22103, 39990, 106175)	18	(465, 2015, 9548, 23715, 106175)
$A^{(3)}$	8	\mathfrak{T}_1 : (228, 0, 0, ...) \mathfrak{T}_2 : (0, 76, 0, 76, 684, 532, 152, 532, 1520)	9	(76, 0, 0, 0, 76, 76, 304, 1520)
$A^{(4)}$	8	\mathfrak{T}_1 : (184, 598, 1242, 391, 0, 0) \mathfrak{T}_2 : (0, 0, 0, 69, 23, 0, 23, 46, 161, 391, 1012, 2300, 5796)	15	(46, 161, 391, 897, 2093, 5796)
$A^{(5)}$	7	\mathfrak{T}_1 : (106, 0, 0, 53, 901, 3233, 954, 53, 0, 0, ...) \mathfrak{T}_2 : (0, 0, 0, 0, 0, 0, 106, 265, 106, 636, 689, 477, 583, 371, 1325, 2915, 5830, 9964)	14	(53, 0, 0, 0, 0, 53, 106, 583, 1484, 3922, 9964)

two were considered in [5]) and satisfies the conditions of [5, Thm. 2]. In particular, there are at most $|V_S| - 2 = 2$ nonzero-valued variable nodes which are both in V_S (V_S is a minimal stopping set contained in V); and there is at least one zero-valued measurement node among the neighbors of V_S . However, it can be readily seen that the IPA will output $\hat{x} = (0, 0, 1, 0, 0, 0)$, i.e., it recovers only one nonzero variable node (v_4 and v_5 are both connected to c_2 and c_4 and thus indistinguishable; hence, the IPA will definitely fail). We believe that the main problematic issue in the proof given in [5] is that variable nodes outside of the minimal stopping set V_S are not considered. Despite the fact that such nodes will be recovered as zeros in the end (because of the specific implementation of the IPA, see Line 20), during iterations they still can “disturb” the values inside of the stopping set.

Finally, we remark that since the statement of [5, Thm. 2] is used in the proof of [5, Thm. 3], it should be further verified.

V. HEURISTIC TO FIND SMALL-SIZE TERMATIKO SETS

As shown in Section III, (small-size) stopping sets may contain termatiko sets as proper subsets. Thus, one way to locate termatiko sets is to first enumerate all stopping sets of size at most τ (for a given binary measurement matrix and threshold τ) and then look for subsets that are termatiko sets. For a given binary measurement matrix A , small-size stopping sets can be identified using the algorithm from [10], [11].

VI. NUMERICAL RESULTS

In this section, we present numerical results for different specific measurement matrices and also for ensembles of measurement matrices. For all matrices we first find all stopping sets of size less than some threshold using the algorithm from [10], [11]. Then, we exhaustively search for termatiko sets as subsets of these stopping sets as explained in Section V. The results are tabulated in Table I for five different measurement matrices, denoted by $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, $A^{(4)}$, and $A^{(5)}$, respectively. Due to the heuristic nature of the approach, the estimated termatiko distance is a true upper bound on the actual termatiko distance, while the estimated multiplicities are

true lower bounds on the actual multiplicities. Measurement matrix $A^{(1)}$ is a 33×121 parity-check matrix of an array-based LDPC code of column weight 3 and row weight 11 [12], $A^{(2)}$ is the parity-check matrix of the $(155, 64)$ Tanner code from [13], $A^{(3)}$ is taken from the IEEE802.16e standard (it is the parity-check matrix of a rate-3/4, length-1824 LDPC code; using base model matrix A and the alternative construction, see [10, Eq. (1)]), $A^{(4)}$ is a 276×552 parity-check matrix of an irregular LDPC code, while $A^{(5)}$ is a 159×265 parity-check matrix of a $(3, 5)$ -regular LDPC code built from arrays of permutation matrices from Latin squares. For the matrix $A^{(1)}$, we have also compared the results with an exact enumeration of all termatiko sets of size at most 5. When considering all stopping sets of size at most 11, the heuristic approach finds the exact multiplicities for sizes 3 and 4, but it underestimates the number of termatiko sets of size 5 by about 7.5% (the missing ones are subsets of stopping sets of size 12 to 14), which indicates that higher order terms (for all tabulated matrices) are mostly likely strict lower bounds on the exact multiplicities. As can be seen from the table, for all matrices except $A^{(3)}$, the estimated termatiko distance is about half the stopping distance. Also, the smallest-size termatiko sets all correspond to termatiko sets with all measurement nodes in N connected to both T and S (cf. Theorem 1). Note that the matrix $A^{(1)}$ is from a family of array-based column-weight 3 matrices, parametrized by an odd prime p . In the general case, the number of columns is p^2 , while the number of rows is $3p$ [12]. It is known that the minimum distance (the measurement matrix is regarded as the parity-check matrix of an LDPC code) for $p \geq 5$ is 6 [14, Thm. 3]. Using the specific structure of the *support matrix* of codewords of weight 6 (see [14, Thm. 4]), it can be shown that there always exist termatiko sets of size 3 for $p \geq 5$, and also that this is the smallest possible size. Thus, the family of parity-check matrices of array-based LDPC codes of column weight 3 is an example of a family of measurement matrices in which the termatiko distance is exactly half the minimum distance.

Now, consider the protograph-based $(3, 6)$ -regular LDPC

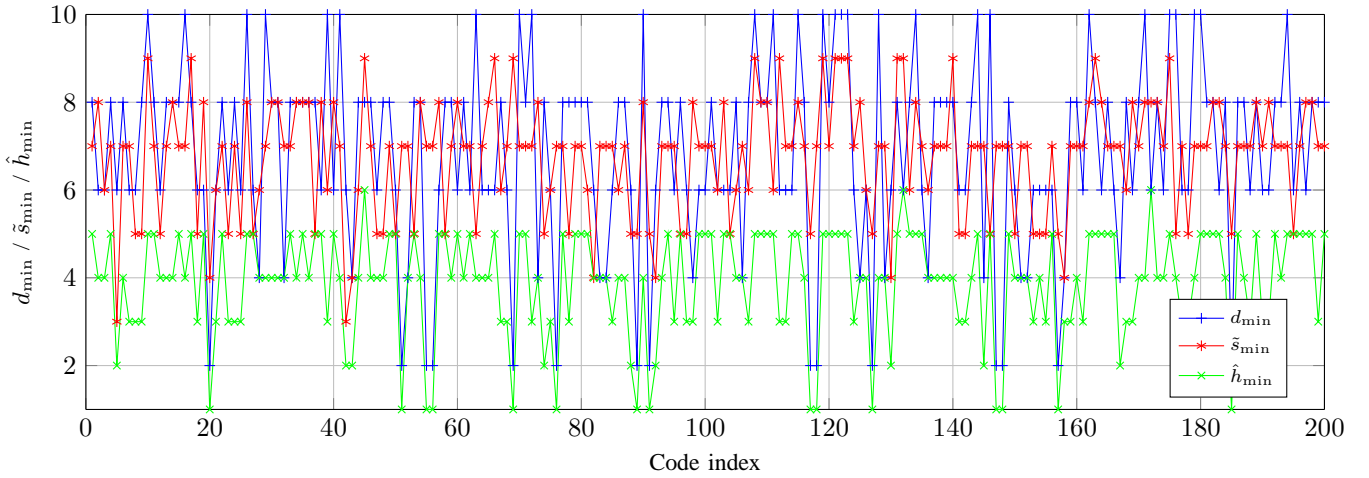


Fig. 5. Minimum distance d_{\min} , minimum size of a noncodeword stopping set \tilde{s}_{\min} , and estimated termatiko distance \hat{h}_{\min} versus code index for randomly generated binary measurement matrices from a protograph-based (3, 6)-regular LDPC code ensemble.

code ensemble defined by the *protomatrix* $\mathbf{H} = (3, 3)$. We randomly generated 200 parity-check matrices from this ensemble using a lifting factor of 100 (the two nonzero entries in the protomatrix are replaced by random row-weight 3 *circulants* (each row is a right-shift of the row above it) of size 100×100). For each lifted matrix, we first found all stopping sets of size at most 16 using the algorithm from [10], [11]. Then, the termatiko distance was estimated for each matrix as explained above. The results are depicted in Fig. 5 as a function of the code index (the blue curve shows the minimum distance d_{\min} , the red curve shows the minimum size of a noncodeword stopping set, denoted by \tilde{s}_{\min} , while the green curve shows the estimated termatiko distance \hat{h}_{\min}). The average d_{\min} , s_{\min} , and \hat{h}_{\min} (over the 200 matrices) are 6.84, 5.92, and 3.90, respectively. We repeated a similar experiment using a lifting factor of 200 in which case the average d_{\min} , s_{\min} , and \hat{h}_{\min} (again over 200 randomly generated matrices) became 9.21, 7.75, and 5.80, respectively.

We remark that similar results (not included here) as the ones depicted in Fig. 5 have been obtained for other ensembles of measurement matrices as well. For instance, both a family of irregular rate-1/2 LDPC codes (1000 codes from the family have been considered) and the rate-1/2 accumulate repeat jagged accumulate ensemble from [15] show similar behaviour.

VII. CONCLUSION

In this work, we have introduced termatiko sets and shown that the IPA fails to fully recover a nonnegative real signal $\mathbf{x} \in \mathbb{R}_{\geq 0}^n$ if and only if the support of \mathbf{x} contains a nonempty termatiko set, thus giving a complete (graph-theoretic) description of the failing sets of the IPA. An extensive numerical study was presented showing that having a termatiko distance strictly smaller than the stopping distance is not uncommon. In some cases, the termatiko distance can be as low as half the stopping distance. Thus, a measurement matrix (for the IPA) should be designed to avoid small-size termatiko sets, which is considered as future work.

REFERENCES

- [1] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [2] —, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
- [3] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [4] F. Zhang and H. D. Pfister, "Verification decoding of high-rate LDPC codes with applications in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5042–5058, Aug. 2012.
- [5] V. Ravanmehr, L. Danjean, B. Vasić, and D. Declercq, "Interval-passing algorithm for non-negative measurement matrices: Performance and reconstruction analysis," *IEEE J. Emerging Sel. Topics Circuits Systems*, vol. 2, no. 3, pp. 424–432, Sep. 2012.
- [6] V. Chandar, D. Shah, and G. W. Wornell, "A simple message-passing algorithm for compressed sensing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 1968–1972.
- [7] H. V. Pham, W. Dai, and O. Milenkovic, "Sublinear compressive sensing reconstruction via belief propagation decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, Korea, Jun./Jul. 2009, pp. 674–678.
- [8] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7434–7464, Nov. 2013.
- [9] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.
- [10] E. Rosnes, Ø. Ytrehus, M. A. Ambroz, and M. Tomlinson, "Addendum to "An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices"," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 164–171, Jan. 2012.
- [11] E. Rosnes and Ø. Ytrehus, "An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4167–4178, Sep. 2009.
- [12] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. Turbo Codes & Rel. Topics*, Brest, France, Sep. 2000, pp. 543–546.
- [13] R. M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," in *Proc. Int. Symp. Commun. Theory and Appl. (ISCTA)*, Ambleside, England, Jul. 2001.
- [14] K. Yang and T. Hellesteth, "On the minimum distance of array codes as LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3268–3271, Dec. 2003.
- [15] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 3, St. Louis, MO, Nov./Dec. 2005, pp. 1152–1156.